Using Deception to Hide Things from Hackers: Processes, Principles, and Techniques

Jim Yuill[1], Dorothy Denning[2], and Fred Feer[3]

[1]*Computer Science Department*
North Carolina State University, USA,
E-mail: jimyuill@pobox.com

[2]*Department of Defense Analysis*
Naval Postgraduate School, USA,
E-mail: dedennin@nps.edu

[3]*U.S. Army, CIA, RAND, ret.*
E-mail: ffeer@earthwave.net

Abstract:

Deception offers one means of hiding things from an adversary. This paper introduces a model for understanding, comparing, and developing methods of deceptive hiding. The model characterizes deceptive hiding in terms of how it defeats the underlying processes that an adversary uses to discover the hidden thing. An adversary's process of discovery can take three forms: direct observation (sensing and recognizing), investigation (evidence collection and hypothesis formation), and learning from other people or agents. Deceptive hiding works by defeating one or more elements of these processes. The model is applied to computer security, and it is also applicable to other domains.

Keywords: *computer security, hiding, denial, deception, operations security*

Introduction

Hiding things from hackers is common practice in computer security. Routinely, systems and files are hidden behind firewalls and access-controls, and data are hidden with encryption. These common forms of hiding typically work by denying information to hackers. Another way to hide things is by using deception. Currently, deception is an emerging and promising means for computer security, as seen with honeypots (Spitzner, 2003). This paper examines the use of deception as a means of hiding things from hackers.

Deceptive hiding can be used in a wide variety of computer security applications. One such application involves hiding information about a network's topology, vulnerabilities, and assets from hacker reconnaissance (for example, scanning). The honeypot *honeyd* for example, intercepts connections to unused network addresses and impersonates computers at those addresses (Spitzner, 2003). Its ruse makes it difficult for hackers to find real computers and to scan the network without being detected.

Deception can be used to hide computer-security devices, including firewalls, intrusion detection systems, keystroke loggers and honeypots. For example, a firewall can send fake ICMP 'host unreachable' messages in response to disallowed packets, making it appear that the firewall, and victim computers behind it, are not on the network.

ISSN 1445-3312 print/ISSN 1445-3347 online

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2006		**00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Using Deception to Hide Things from Hackers: Processes, Principles, and Techniques	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School,Center of Terrorism and Irregular Warfare,Monterey,CA,93943**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Deception offers one means of hiding things from an adversary. This paper introduces a model for understanding, comparing, and developing methods of deceptive hiding. The model characterizes deceptive hiding in terms of how it defeats the underlying processes that an adversary uses to discover the hidden thing. An adversary?s process of discovery can take three forms: direct observation (sensing and recognizing), investigation (evidence collection and hypothesis formation), and learning from other people or agents. Deceptive hiding works by defeating one or more elements of these processes. The model is applied to computer security, and it is also applicable to other domains.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**15**	

Computer security deception is defined as the actions taken to deliberately mislead hackers and to thereby cause them to take (or not take) specific actions that aid computer security (JDD, 1996). Often, for deceptive hiding, the objective is to cause the hacker to not take a particular action, such as accessing a server.

Furthermore, computer security deception aims to mislead a hacker into a predictable course of action or inaction that can be exploited or otherwise used to advantage (Dewar, 1989). In general, actions that cause the hacker to act dangerously or unpredictably should be avoided. For example, suppose a system administrator hides network logs to prevent hackers from erasing their tracks. If the expected logs are not found, a hacker may erase the entire hard drive, just to be safe. An important aspect of deception planning, therefore, is anticipating such unintended consequences and taking actions to mitigate their effect.

In the context of computer security, things are hidden from an agent, human or computer. The agent whom the thing is hidden from will be referred to as the *target*. The target is a hacker or a hacker's automated agent (for example, a worm). For deception operations, in general, the adversary who is being deceived is referred to as the *deception target*. For deceptive hiding, the target of hiding is also the deception target.

This paper explains how deceptive hiding works in terms of how it misleads, or tricks, a particular target (hacker). However, the deception planner's ultimate purpose is not misleading the target, but improving computer security in some specific way. Deception's trickery can be both alluring and intriguing, making it is easy to lose sight of the deception's ultimate purpose.

The paper describes deceptive hiding through a process model. The model's purpose is to provide a framework for understanding, comparing, and developing methods of deceptive hiding. Although the model is based on general principles and techniques that are domain-independent, the paper focuses on the model's application to computer security. The goal is to help the security professional evaluate, compare, configure, and use existing deceptive hiding techniques (for example, honeyd); and to help explore possibilities when creating new techniques.

The model characterizes methods of deceptive hiding in terms of how they defeat the underlying processes that a target uses to discover the hidden thing. This process is decomposed into three means of discovery: direct observation (sensing and recognizing), investigation (evidence collection and hypothesis formation), and learning from other people or agents. Although the focus is on deceptive hiding, many of the concepts are also relevant to non-deceptive hiding.

The next section introduces the process of deceptive hiding. Subsequent sections describe the three means of discovery and how they are defeated; a final section concludes.

The Process of Deceptive Hiding

Deception has two aspects, hiding and showing. This section first reviews these aspects of deception, and also, the earlier work on deception. It then discusses how deceptive hiding works and the processes involved.

An overview of deception

Deception is a form of perception in which a target is intentionally led to an incorrect perception, through the actions of another (Whaley, 1982) Deception is distinguished from unintentional acts of misrepresentation and from self-induced acts of misrepresentation (self-deception).

Bell and Whaley categorize deceptions as hiding and showing (Bell and Whaley 1982, and Whaley 1982). *Deceptive hiding* conceals or obscures a thing's existence or its attributes in a way that intentionally misleads the target. It is distinguished from *denial*, which may also involve hiding, but without the intent to mislead. Denial simply withholds information from the target. Encryption, which overtly conceals a message but not its existence, is an example. Steganography, on the other hand, which aims to hide the existence of a communication, is deceptive, as it uses a misleading data carrier (for example, text is hidden in the low-order bits of an image file in such manner that the text is not visible to the naked eye).

Deceptive showing makes something that does not exist appear as if it does by portraying one or more of its attributes. For example, after several unsuccessful logins, a computer can continue to prompt for passwords, but ignore them and not permit login. The computer is deceptively showing login prompts.

Hiding and showing are both present in any act of deception (Bell and Whaley, 1982). When showing the false, the truth must also be hidden. When something is hidden, something else is shown instead, even if only implicitly. Further, deceptions are often constructed of multiple ruses, employing both hiding and showing. For example, a honeypot can deceptively impersonate (that is, show) a network server, while deceptively hiding a keystroke logger. When a deception uses both hiding and showing, the deception may be characterized as hiding or showing, according to the planner's primary intent. For instance, a server's banner is modified to display a false model and version number. The banner is showing falsehood, but the primary intent is hiding the server's true model and version from hackers and worms.

Bell and Whaley offer a taxonomy of deceptive techniques based on three ways of hiding: masking, repackaging, and dazzling; and three ways of showing: mimicking, inventing, and decoying (Bell and Whaley, 1982). The taxonomy has been used in both the military and computer security literature (USMC 1989, Julian 2002). The military deception literature also lists common types of battlefield deceptions, examples being camouflage, feints (fake attack-initiation), ruses (tricks designed to deceive), demonstrations (fake force deployment), and displays (the showing of fake military forces or equipment, for example, inflatable tanks) (U.S. Army 1998, Dewar 1989, Fowler and Nesbit 1995). Cohen (1998) and Rowe and Rothstein (2004) have shown how these can be applied to computer network defense. Rowe and Rothstein also give a taxonomy of deception techniques based on semantic cases in computational linguistics such as agent, instrument, location-from, time-at, and purpose. In addition, Rowe has developed a taxonomy for deception in virtual communities (Rowe, 2005). The taxonomy applies primarily to computer misuse, and not to computer security.

The model presented in this paper extends this earlier work by showing how deceptive hiding can be understood in terms of processes, mainly the discovery processes used by a target to acquire information. Particular hiding techniques work by defeating elements of these processes.

An overview of deceptive hiding

Hiding keeps the target from knowing about the hidden thing's existence or its attributes. As a result, the target will be unaware of the thing, certain it does not exist, uncertain of its existence, or left with incomplete or inaccurate information about it. Hiding can prevent discovery of the hidden thing, or it can make discovery more difficult or time consuming.

There are three different ways a target can discover a particular thing:

1) direct observation of the thing,
2) investigation based on evidence of the thing, and
3) learning about the thing from other people or agents.

These three means of discovery comprise the target's *discovery process*. Hiding works by defeating this process, which is driven by two elements: capabilities and a course of action. The target's *discovery capabilities* are defined as the resources, skills, and abilities that the target has for discovery. The *discovery course-of-action* is the way the target carries out the discovery process; it includes how, when and where the target looks for things. This suggests that the target's discovery process can be defeated by affecting either the target's capabilities or the target's course of action. For instance, installing a firewall can ensure a hacker's port scan is not capable of directly observing a computer's servers. Alternatively, deploying an enticing honeypot could divert the hacker's course-of-action so that the port-scans reveal the honeypot rather than the hidden servers.

It is assumed that the target intends to discover the hidden thing. Another way to hide is to affect the target's intentions. For example, to deter network scanning, a company could fire any employee found scanning its intranet. Hiding by altering intentions is not addressed by this paper.

The three discovery processes are now examined in terms of how they work and how they can be defeated through deceptive hiding.

Direct Observation

When hacking a network, much of what the hacker knows about the network is learned by direct observation. For example, a port scan allows the hacker to observe a network's computers and servers. After gaining access to a computer, the hacker can use system utilities to observe the computer's resources, such as files, programs, and running processes; application programs to observe business and user data; and network clients to observe servers and their contents.

After the discovery process is described, hiding is examined to show how it defeats that process.

The discovery process for direct observation

The discovery process for direct observation involves *sensing* and *recognizing*. The process is illustrated in Figure 1 and explained here. The deception target's human sensors (for example, eyes) are used to observe. The target may also rely upon one or more external sensors, such as a network port scanner or packet sniffer. Information flows to and from the sensors over media (for example, network cables, routers, and computer monitors). The hidden thing is observed within the environment in which it resides (for example, a private computer network). After the target receives the sensory input, recognition occurs within the target's brain. Recognition is a cognitive process involving the target's knowledge and

understanding. Discovery occurs when the hidden thing is identified (that is, recognized) based on expected patterns.

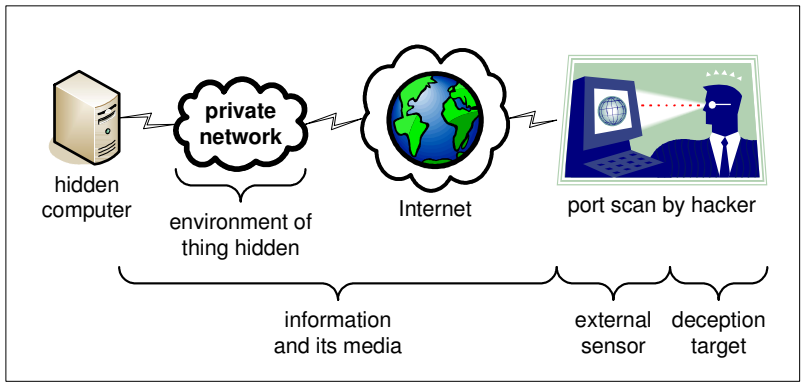

Figure 1 : The process of direct observation, illustrated by a computer-security example

A sensor receives information and then provides images to the target. These images can be conveyed to the target in a variety of ways. For instance, when a target's eyes are used to observe a computer, the image is conveyed visually. When the target observes the computer by using a port scanner as a sensor, the image is conveyed descriptively via text. Typically, sensors work in a deterministic manner, and their operation is based on mechanisms such as software and electronics (for example, the port scanner), or physiology (for example, eyes). Recognition, on the other hand, is much less deterministic than the sensors. The target might miss identifying something even if it is seen, especially if the target does not know what patterns to look for. Recognition depends on knowledge and intelligence, real or artificial.

The target's sensor and recognition capabilities are considered to be distinct elements in the model. In practice, however, both capabilities may be present in a single device. A network intrusion-detection system (NIDS), for example, can have a sensory component consisting of a packet sniffer and a recognition component based on matching packet information against attack signatures or statistical anomalies.

The target can discover things by actively searching for them or through passive observation. Discovery involves bringing the sensors to bear upon the hidden thing. The hidden thing is then distinguished and recognized from within the environment in which it resides.

How hiding defeats direct observation

Hiding defeats direct observation by defeating the targets sensor(s) and/or recognition. The *sensor* is defeated if it does not provide the target with a distinguishable image of the hidden thing. For example, when steganography is used to hide text within a picture, the target's sensors (graphics browser and eyes) cannot distinguish the text data.

Recall that the target's discovery process can be defeated through the target's 1) discovery capabilities or 2) course of action. For direct observation, this means preventing the target's sensor capabilities, or the way the sensor is used, from providing a distinguishable image of the hidden thing. One way to achieve this is by altering an element of the discovery process that is external to the target and the target's sensors. Such elements include the hidden thing's location, appearance or environment, or the information flows to the sensor. For example, placing a firewall between a server and the Internet would alter the information flows between the server (hidden thing) and the hacker's port scanner (sensor), thereby defeating the

scanner's capabilities. Alternatively, the hacker's use of the scanner could be defeated by altering the server's location; for example, the server could be placed on a subnet that the hacker is not likely to scan.

Hiding can also be achieved by taking direct action against the target's sensor capabilities or the target's use of the sensor. For example, launching a denial of service attack against the hacker's computer during a port scan could impair use of the port scanner.

Table 1 : Hiding techniques that defeat the target's sensors

Action Type	***Ways to Defeat Sensor*** (sensor does not provide a distinguishable image of the hidden thing)
alter location of hidden thing	place the hidden thing where the target is not likely to observe: • place critical files in obscure directories place the hidden thing where the target's sensors cannot observe: • hide laptop behind NAT (network address translation) device • hide information within a cover medium, using steganography
alter appearance of hidden thing	make the hidden thing not reflect information to sensor: • computer eludes ping scans by not replying to pings make the hidden thing blend in with background: • password file given non-descriptive name, to elude hackers' automated searches for files named 'pass*' alter the hidden thing's appearance, so the target's sensor is not capable of observing it • encrypt message (the target can observe the cipher text, but not the plain text)
alter environment of hidden thing	create noise in environment: • add bogus files to make it harder to find critical ones alter components in environment to prevent access to the hidden thing: • hide network data from sniffers by replacing Ethernet hubs with switches
alter information flows to sensor	alter information needed by sensor: • router drops incoming pings to hide its network's computers from ping scans • delay responses to login attempts so hacker does not have time to guess password add components to communication path • firewall added to prevent certain flows to or from computers on network
diminish target's sensor capabilities	disable or degrade the sensor: • perform a DoS attack against a hacker's port-scanner reduce the target's time available for observation • quickly detect and stop target's reconnaissance, such as port scans
misdirect target's use of sensor	cause the target to observe at the wrong place or time • create a diversion for the hacker

Table 1 summarizes and illustrates the options for defeating sensors. The first column lists the general types of actions outlined above, while the second gives greater specificity and examples. (Subsequent tables in the paper will follow this format.) The table provides the deception planner with a framework for evaluating and developing hiding techniques. The

action-types listed in the first column are intended to be exhaustive and mutually exclusive. The body of the table presents a broad, though not exhaustive, collection of common hiding techniques for deception and denial. Some hiding techniques affect multiple elements of the discovery process, so they could be placed in multiple tables or categories within a table.

The target's *recognition process* attempts to identify the hidden thing from among the images provided by sensors. Assuming the sensors provide a distinguishable image of the hidden thing, recognition is defeated if the target is not able to identify the hidden thing from the sensory input. For instance, to hide a virtual private network (VPN) server on a demilitarized zone (DMZ), three honeypot VPN servers could be added to the DMZ. A hacker's port scan reveals all four VPN servers, but the hacker is unable to recognize which is real.

Table 2 : Hiding techniques that defeat the target's recognition

Action Type	***Ways to Defeat Recognition*** (the hidden thing cannot be identified in the sensor's images)
alter location of hidden thing	locate where the target observes, but does not expect the hidden thing: • put sensitive document files in a software application's directory
alter appearance of hidden thing	disguise the hidden thing by making it mimic something expected in environment: • use ports that make a server appear like a workstation to scanners make the hidden thing appear as something the target does not recognize: • use unconventional names for sensitive files
alter environment of hidden thing	make things in the environment resemble the hidden thing: • place a highly valuable workstation on a LAN with many workstations that have low value, but that appear the same to hackers' scans
alter information flows to sensor	generate false information that is received by the sensor, but misleads recognition • *honeyd* thwarts scanning by impersonating computers at unused IP addresses • *nmap*'s decoy port-scan hides the scan's source address by sending many packets with fake source addresses
diminish target's recognition capability	disable or degrade the recognition process: • exhaust the hacker by providing an overwhelming amount of false information reduce target's time available for recognition • stop the hacker before the hacker recognizes critical systems and information prevent target from acquiring the understanding needed to recognize the hidden thing • limit publication of information that could aid hacker
misdirect target's recognition process	cause target to expect something other than the hidden thing • misinform hacker about identity of network elements

The target's recognition process can be defeated through the target's 1) recognition capabilities or 2) course of action. The recognition capabilities are a function of 1) the target's cognitive abilities, skill and experience in identifying the hidden thing from the sensor's image, and 2) the target's available resources, including time. The target's course of action includes how, when and where the target recognizes things, which are all influenced by the target's expectations. For example, a hacker would expect, and more readily recognize, banking-industry security devices on a bank's network than on a typical home network.

Table 2 illustrates how a target's recognition process can be defeated in order to hide. The table's first column is the same as in Table 1. The reason is that recognition is defeated by the same types of actions that are used to defeat sensors. Table 2's second column lists specific hiding techniques applicable to defeating recognition.

Investigation

Investigation is a means of discovery that infers a thing's existence from evidence rather than direct observation. Investigation is used in many domains, for example law enforcement (determining guilt based on evidence) and health care (diagnosing illness from symptoms).

In general, investigation is used to discover a thing that existed in the past when the thing was either not directly observed or a reliable recording of the observation is not available (for example, a computer log, video tape, or witness' testimony). Investigation is also used to discover things that exist in the present, but which cannot be directly observed. Things in the future can be anticipated based on indicators, but cannot be investigated because evidence of them does not exist.

Hackers often use investigation to obtain information about the current state of a victim network's topology, as well as its defences, vulnerabilities, and assets. For example:

- By acquiring a network's computer names, a hacker might be able to deduce which computers are vulnerable (McClure et al., 1999). Computers with names containing 'test' such as 'test-network-gateway,' may be indicative of systems that have not been configured securely.
- A variety of techniques are available for obtaining evidence that reveals firewalls and their access control lists (ACLs) (McClure et al., 1999). Firewalking can reveal which ports are open or blocked by a firewall (Goldsmith and Schiffman, 1998). (Firewalking sends a TCP packet with an IP TTL field set to one hop beyond the firewall. If the reply is the ICMP error message "time to live exceeded in transit", then it is evidence that the TCP port is open.)
- Email sent to a public newsgroup can reveal the internal IP address of a sending computer that is otherwise hidden by a NAT device.

Investigation is an inherent first phase of most network attacks. Deceptive hiding can be used to defeat these and other hacker investigations. When using deceptive hiding for computer security, the hacker is the investigator and deception target. When hiding things from investigation, the investigator is an adversary. Viewing an investigator as an adversary is somewhat unusual, as investigators are normally the 'good guys', for example, policemen and scientists. Of course, when the hacker is hiding things, the cyber cops become the investigators.

The following two sub-sections describe the process of investigation and how that process can be defeated, respectively. The treatment of the investigation process is adapted from David Schum's excellent research on investigation for jurisprudence (Schum, 1999).

The investigation process

Investigation is an iterative process of creating *hypotheses* and acquiring *evidence* about the thing being investigated. Typically, the investigator works with incomplete evidence, so there can be many plausible hypotheses that are consistent with the evidence. At any point during

the process, the investigator can either develop new hypotheses based upon the available evidence or search for new evidence to answer questions relating to the investigator's current evidence and hypotheses. As the investigation unfolds, each piece of new evidence reduces the number of possible hypotheses and inspires the creation of more accurate and detailed hypotheses. New evidence suggests new questions and hypotheses, and these in turn drive the collection of further evidence. The information and understanding obtained is cumulative.

There are two types of hypotheses that the investigator develops and works with: discovery hypotheses and collections hypotheses. *Discovery hypotheses* explain that which is being investigated in terms of available evidence, and they culminate in the recognition or discovery of the hidden thing. *Collections hypotheses* explain where additional evidence might be found, and they guide the investigator's search for new evidence. New evidence can be acquired through direct observation (as described earlier) or from other people or agents (as described later). The collected evidence may include false and irrelevant information that misleads the investigator.

Investigations vary in the amount of evidence collected and hypotheses formed. Some are simple and produce immediate results. For example, after breaking into a computer and detecting evidence of a hidden keystroke logger, a hacker could immediately conclude that the computer is a honeypot. Other investigations are more complex, requiring the investigator to combine multiple pieces of evidence acquired over time. Instead of discovering a keystroke logger, the hacker might observe that it is not possible to create outgoing connections and that the computer contains no user data. By observing these conditions over time and considering them together, the hacker deduces the machine is a honeypot.

The process of investigation requires creativity. It also requires deliberate choices. Investigation comes at a cost, so the investigator cannot follow every hypothesis and seek evidence to answer every possible question. The investigator will be limited by available resources (including time), to collect, process, and retain evidence. How the investigation proceeds will depend upon the investigator's resources and decisions about how they are used. If the choices are bad, the investigator will make false hypotheses, collect the wrong evidence, and waste resources on useless paths of investigation.

Evidence often has a temporary existence, which can pose significant problems during the initial investigation. As time progresses, an increasing amount of evidence will no longer be obtainable. For example, log files are eventually erased or destroyed, and peoples' memory fades. The investigator needs to gather and preserve evidence before the opportunity is lost. However, much useful evidence may not be discernable at the beginning of the investigation. The discernment of evidence requires understanding of the case, and the investigator acquires understanding over time. The investigator can reduce the loss of temporarily-available evidence. By making many hypotheses, and very general hypotheses, the investigator can collect a large amount of evidence that is potentially useful. However, the investigator has limited resources for collecting and storing evidence.

Investigation is a necessary first phase of most network attacks. Further, the investigation process is weakest at the beginning of an investigation, as just described. Thus, a hacker's initial network investigation can be a *critical vulnerability*, and relatively easy for defenders to exploit. (In military theory, a critical vulnerability is a specific type of vulnerability. A combatant's vulnerability is a critical vulnerability if it can be exploited to destroy a capability without which the combatant cannot function effectively (USMC, 1997)).

How hiding defeats investigation

The inherent difficulties of investigation can be exploited through deception. If evidence is hidden, the investigator may form false hypotheses, ask erroneous questions, and pursue futile investigation tracks. The investigator may terminate what would have been a fruitful track. In situations where several pieces of evidence are needed to discover a thing, it may suffice to hide some of the evidence in order to prevent discovery. In situations where evidence has a limited lifetime, it may be enough to interfere with the start of the investigation or delay its progress.

The investigation process is defeated if 1) the target does not recognize the hidden thing, or 2) if the target's recognition is made sufficiently uncertain. This can be accomplished by defeating either of the sub-processes that comprise the investigative process: evidence collection and the creation of discovery hypotheses.

The *evidence collection process* includes 1) the target's creation of collections hypotheses and 2) the target's acquisition of information. This process is defeated by preventing the target from obtaining the evidence needed for recognition. Two types of actions can be taken to defeat the target's evidence collection: 1) alter the evidence available in the environment; that is, do not create evidence, hide evidence, or destroy evidence, and 2) weaken the target's evidence-collection process by diminishing the target's capabilities or by misdirecting the target's actions. See Table 3.

The target's evidence collection can be defeated more effectively if the target's search for evidence can be anticipated. There are two common searches for evidence that are especially vulnerable. The first are superficial searches, which result when many things must be examined, and time limitations prohibit a thorough examination. For example, a hacker's network scan may involve examining thousands of computers. To speed up the process, hackers often first perform a superficial ping scan to locate running computers. They then perform a port scan on the running computers. Such superficial examinations can be very vulnerable to deception. Second are predictable searches for evidence performed by computer programs. These searches lack human intelligence. For instance, hackers use open-source vulnerability scanners, and these scanners look for specific types of evidence. Hiding evidence from popular hacker tools can defeat a large portion of the hacker investigations on a network.

The other way to hide from investigation is by defeating the target's creation of discovery hypotheses. However, it is only necessary when the target is able to obtain the evidence needed for recognition. Hiding is accomplished by preventing the target from creating the discovery hypotheses needed for recognition. There are two ways to defeat the creation of discovery hypotheses: 1) ensure the target is not capable of creating the necessary discovery hypotheses, and 2) ensure the target's process of creating discovery hypotheses does not lead the target to recognize the hidden thing. Table 4 elaborates this.

Learning from Other People or Agents

The third way a target can discover something is to learn about it from another entity. This section describes the learning process and how it can be defeated.

Table 3 : Hiding techniques that defeat the target's evidence collection

Action Type	***Ways to Defeat Evidence Collection*** (the necessary evidence is not collected)
block evidence creation	find a way to do things so evidence is not created: • configure outgoing mail server to remove sender's IP address from mail headers
hide evidence	hide evidence that could be acquired by direct observation or learned from other people or agents
destroy evidence	destroy evidence before the target can collect it, either at once or by entropy over time • remove sensitive information from memory and disk after use
diminish target's evidence-collection capabilities	reduce the target's time available for collection • quickly detect and abort hackers before they find critical information • delay the target's evidence collection, so that it exceeds the target's available time
misdirect target's evidence-collection	misdirect the target's collection activities, to keep the target away from necessary evidence; for example, create false evidence that causes the target to look for evidence in the wrong places confuse the target, so the target cannot form the collection or discovery hypotheses needed to obtain necessary evidence; for example, create false evidence that contradicts real evidence reduce the target's perceived reliability of necessary evidence; for example, create false evidence that is of the same type as the real necessary evidence, and allow the target to learn that false evidence has been created

Table 4 : Hiding techniques that defeat the target's creation of discovery hypotheses

Action Type	***Ways to Defeat the Creation of Discovery Hypotheses*** (even if the target has the necessary evidence, the target cannot create the necessary discovery hypotheses)
diminish target's capabilities for creating discovery hypotheses	cause target's capabilities to be insufficient; for example, reduce target's available time
misdirect target's creation of discovery hypotheses	mislead target; for example, create false evidence, or hide true evidence, and thereby cause the target to form incorrect discovery hypotheses confuse the target, so the target cannot form the necessary discovery hypotheses; for example, create false evidence that contradicts real evidence

The learning process

The learning process is a discovery process wherein the target learns of the hidden thing from a *discovery agent*. The discovery agent can be a person or a device with sensor and recognition capabilities, such as a software agent. The agent discovers the hidden thing through its own discovery process, which can be direct observation, investigation, or learning. The agent then reports the discovery, and the report is communicated to the target. The report can be sent directly to the target (for example, via an email), or recorded and placed somewhere accessible to the target (for example, a website). The discovery agent may act autonomously or under the direction of the deception planner or the target. Figure 2 illustrates.

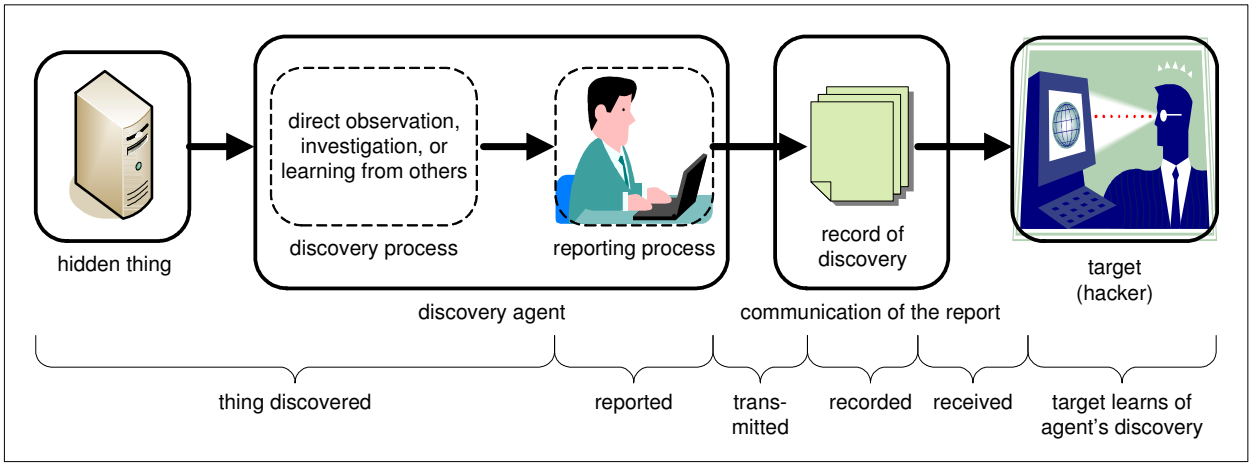

Figure 2 : How the target learns from other's, or agents', discoveries

In practice, the target may learn of a thing through a series of agents. For example, the target learns of the thing from person A, who learned of it from person B, and so on, the first person having acquired it from direct observation or investigation.

Hackers acquire much of their knowledge from others. For instance, through footprinting they learn about a victim's network from publicly available information (McClure et al., 1999). Typical sources include DNS servers, which record the IP addresses and domain names of computers on a network, and company websites, which may contain information about the company's networks. Hackers also learn through distribution lists, chat channels, and other online forums.

How hiding defeats learning

Hiding defeats the learning process by defeating the discovery agent, communication of the report, or the target's recognition. The *discovery agent* is defeated if it does not discover the hidden thing or does not attempt to report it. The *communication of the report* is defeated if the report is not successfully transmitted, recorded, or received by the target (assuming the discovery agent has attempted to communicate the report). The *target's recognition* is defeated if the target does not learn of the hidden thing from the report (assuming the target has received the report). Table 5 elaborates this.

Conclusions

This paper explains deceptive hiding in terms of defeating the target's discovery process. The model includes three means of discovery: direct observation (sensing and recognizing), investigation (evidence collection and hypothesis formation), and learning from other people or agents (discovery by an agent, report communication, and target recognition); for each hiding defeats one or more of the components of the discovery process. This is accomplished by ensuring that: 1) the target is not capable of discovering the hidden thing, or that 2) the target's course-of-action does not lead the target to discover the hidden thing.

The process model offers a conceptual framework for developing new deceptive hiding techniques and for evaluating existing techniques. The model also offers a common frame of reference for collaboration among security professionals. When hiding a particular thing, the deception planner can determine which discovery methods the target is likely to use. For each method, the tables of hiding techniques can be used to consider the possible ways to hide.

Table 5 : Techniques for hiding when the target learns from other's, or agents', discoveries

Action Type	***Ways to Defeat the Discovery Agent*** (the hidden thing is not discovered and reported)
hide thing from discovery agent	hide thing from the agent's direct observation • give unused addresses on a network fake names to hide real computer-names in reverse DNS lookups. hide thing from the agent's investigation
alter discovery agent's reporting process	instruct discovery agents under control of deception planner to omit hidden thing from reports • omit high-valued assets from published network diagrams • omit sensitive network information on public technical-support forums
diminish discovery agent's capabilities for serving target	cause discovery agent to not serve target: • bribe or 'turn' hackers who serve as discovery agents for others • detect and remove a hacker's network sniffers (discovery agents) degrade capabilities of discovery agents: • modify a hacker's sniffers so they garble captured data. The hacker may regard them as too problematic to use on the network. interfere with target's directions to the discovery agent: • install a firewall to block a hacker's access to an installed sniffer
Action Type	***Ways to Defeat Communication of the Report*** (the hidden thing is not successfully communicated)
alter transmission or receipt of report	block the transmission or receipt of the report • configure firewall to drop outgoing ICMP packets, which are used by the hacker tool LOKI to communicate covertly
alter recorded report	falsify or destroy the recorded report • when a hacker's vulnerability scanner (discovery agent) is found running on a computer inside a network, falsify or erase the recorded results.
Action Type	***Ways to Defeat the Target's Recognition*** (the target does not learn of the hidden thing from the report)
affect report	confuse target by causing discovery agent to report things resembling hidden thing • honeyd impersonates many vulnerable computers, causing a hacker's vulnerability scanner to return an overwhelming number of false positives.
diminish target's learning capability	cause the target's learning resources to be insufficient • reduce the target's time available for the report; for example, law enforcement's aggressive pursuit of a hacker causes the hacker to spend more time on evasion and defence, and thus the target has less time for learning about victims' networks.

The hiding model is applicable to both deceptive hiding and non-deceptive hiding (that is, denial). Non-deceptive hiding defeats the target's discovery process, but without misleading the target.

A topic for future research is extending the discovery-process models to deceptive showing. In this case, the discovery process would be manipulated to portray something false. The model might also be extended to deceptions aimed at altering the target's intentions, so that the target no longer attempts to discover the hidden thing. Another topic for future research is developing metrics for evaluating the effectiveness of techniques for hiding (and showing). For computer security, the metrics could be based on those used to evaluate other types of security mechanisms.

References

Bell, J., Whaley, B. (1982) *Cheating and Deception*, Transaction Publishers, New Brunswick, NJ.

Cohen, F. (1998) A Note on the Role of Deception in Information Protection, *Computers & Security*, **17**:483-506.

Dewar, M. (1989) *The Art of Deception in Warfare*, David & Charles, London.

Fowler, C., Nesbit, R. (1995) Tactical Deception in Air-Land Warfare, *Journal of Electronic Defense*, **18**(6):37-44.

Goldsmith, D., Schiffman, M. (1998) Firewalking : A Traceroute-Like Analysis of IP Packet Responses to Determine Gateway Access, URL: http://www.packetfactory.net/projects/firewalk [Accessed: August 30, 2006].

Joint Doctrine Division (1996) *Joint Pub 3-58, Joint Doctrine for Military Deception*, Joint Education and Doctrine Division.

Julian, D. (2002) *Delaying-Type Responses for Use by Software Decoys,* master's degree thesis, Naval Postgraduate School, Monterey, CA.

McClure, S., Scambray, J., and Kurtz, G. (1999) *Hacking Exposed : Network Security Secrets and Solutions*, Osborne/McGraw-Hill, Berkeley.

Rowe, N. (2005) Types of Online Deception, *Encyclopedia of Virtual Communities and Technologies*, Idea Group, Hershey, PA.

Rowe, N. and Rothstein, H. (2004) Two Taxonomies of Deception for Attacks on Information Systems, *Journal of Information Warfare,* **3**(2):28 – 40.

Schum, D. (1999) Marshaling Thoughts and Evidence During Fact Investigation, *South Texas Law Review*, **40**(2): 401-454.

Spitzner, L. (2003) *Honeypots : Tracking Hackers*, Addison Wesley, Boston.

U.S. Army (1988) *FM 90-2 Battlefield Deception*, U.S. Army, Washington.

USMC (1989) *FM 15-6 Strategic and Operational Military Deception: U.S. Marines and the Next Twenty Years*, U.S. Marine Corps, Washington.

USMC (1997) *MCDP 1-3 Tactics*, U.S. Marine Corps, Washington.

Whaley, B. (1982) Toward a General Theory of Deception, *Journal of Strategic Studies*, **5**(1):178-192.